

ALAGAPPA UNIVERSITY



[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category–I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003

Directorate of Distance Education

BCA

IV - Semester

101 44

LAB: INTERNET AND JAVA PROGRAMMING

Author:

Deepak Gupta, Asst. Professor, G. L. Bajaj Institute of Technology & Management, Greater Noida



Vikas® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.
E-28, Sector-8, Noida - 201301 (UP)
Phone: 0120-4078900 • Fax: 0120-4078999
Regd. Office: A-27, 2nd Floor, Mohan Co-operative Industrial Estate, New Delhi 1100 44
• Website: www.vikaspublishing.com • Email: helpline@vikaspublishing.com

Work Order No. AU/DDE/DE12/Printing of Course Material/2020 Dated 06.05.2020 Copies - 200

LAB: INTERNET AND JAVA PROGRAMMING

SYLLABI

BLOCK 1: JAVA FUNDAMENTAL PROBLEMS
Simple Java Problems
Class and Objects
Conditional Control using Java
Looping using Java

BLOCK 2: OOP CONCEPTS
Function Overloading Programs
Operator Overloading Programs
Inheritance Programs, Packages
Polymorphism Programs, Message Passing Programs

BLOCK 3: THREAD & VIRTUAL FUNCTION
Threads
Virtual Function Programs

BLOCK 4: I/O AND EXCEPTION HANDLING
Exception Handling Programs
I/O Manipulation Programs

BLOCK 5: NETWORK PROGRAMMING
Applet Programs
Implementation of Simple Network Programs using Java

INTRODUCTION

Java is a class-based, Object-Oriented Programming (OOP) language that is specifically designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers Write Once, Run Anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java Virtual Machine (JVM) regardless of the underlying computer architecture.

Java is strongly associated with the Internet. Internet helped Java to bring it to the forefront and Java influenced Internet by simplifying the web programming and inventing Applet programming. Applets expanded the scope of Internet. Java also addressed two other important issues of Internet, which are security and portability. Internet users can also setup their websites containing Java Applets that could be used by other remote users of the Internet. This feature made Java most popular programming language for the Internet.

This lab manual, *Internet and Java Programming*, contains several programs based on Java concepts, such as class and objects, conditional control using Java, looping using Java, function overloading, operator overloading, inheritance, packages, polymorphism, threads, exception handling, I/O manipulation, Applets and simple network programs using Java. In addition, it will help students in coding and debugging their programs. The manual provides all logical, mathematical and conceptual programs that can help to write programs very easily in Java language. These exercises shall be taken as the base reference during lab activities for students. There are also many Try Yourself Questions provided to students for implementation in the lab.

BLOCK I : JAVA FUNDAMENTAL PROBLEMS

This block contains the programs related to the following concepts:

1. Classes and Objects
2. Conditional Control and Looping in Java

Java is an object-oriented programming language, hence in Java everything is related with classes and objects, along with its attributes and methods. A Class is like an object constructor, or a 'blueprint' for creating objects. In Java, an object is created from a class. You can create multiple objects of one class.

1. Write a Java program for finding the largest of the three numbers.

Program

```
import java.util.*;
class LargeNumFinder
{
    public static void main(String args[])
    {
    int num1, num2, num3, large_num;
        System.out.print("Program for finding the largest
of the three numbers...");
        Scanner scan= new Scanner (System.in);
        System.out.print("Please Enter First Number: ");
        num1=scan.nextInt();
        System.out.print("Please Enter Second Number: ");
        num2=scan.nextInt();
        System.out.print("Please Enter Third Number: ");
        num3=scan.nextInt();
        if(num1>num2 && num1>num3)
           large_num=num1;
        else if(num2>num1 && num2>num3)
           large_num=num2;
        else
           large_num=num3;
        System.out.println("Largest Of The Three Number :
" + large_num);
            }
}
```

Output of the program:

```
Command Prompt

D:\alagappauniv\JavaPrograms>javac LargeNumFinder.java

D:\alagappauniv\JavaPrograms>java LargeNumFinder
Program for finding the largest of the three numbers...
Please Enter First Number: 3453
Please Enter Second Number: 31123
Please Enter Third Number: 656343
Largest Of The Three Number : 656343

D:\alagappauniv\JavaPrograms>
```

2. Write a Java program for finding the factorial of a given number.

Program

```
import java.util.*;
class FactorialOfANumber
      {
public static void main(String args[])
   {
int n, i, fact=1;
      System.out.println("Program for finding the factorial of a number...");
      Scanner scan= new Scanner(System.in);
      System.out.print("Please Enter a No.");
      n=scan.nextInt();
      for(i=n;i>=1;i—)
         {
         fact=fact*i ;
          }
      System.out.println("Factorial of " + n + " is  " + fact);
                        }
                               }
```

Output of the program:

```
Command Prompt

D:\alagappauniv\JavaPrograms>javac FactorialOfANumber.java

D:\alagappauniv\JavaPrograms>java FactorialOfANumber
Program for finding the factorial of a number...
Please Enter a No.10
Factorial of 10 is  3628800

D:\alagappauniv\JavaPrograms>
```

3. Write a Java program for finding the product of digits of input numbers.

NOTES

Program

```
import java.util.*;
class ProductOfDigits
{
    public static void main(String args[])
    {
        int n, res,product=1;
        System.out.println("Program to dispaly the Product
of digits of a number");
        Scanner scan= new Scanner(System.in);
        System.out.print("Please Enter No. =  ");
        n=scan.nextInt();
        System.out.print("Product  of  digits  of  a
number"+""+n+""+ "is = ");
        while(n>0)
        {
            res=n%10;
            n=n/10;
            product=product*res;
        }
        System.out.print(""+product);

    }
}
```

Output of the program:

```
Command Prompt

D:\alagappauniv\JavaPrograms>javac ProductOfDigits.java

D:\alagappauniv\JavaPrograms>java ProductOfDigits
Program to dispaly the Product of digits of a number
Please Enter No. =  532
Product of digits of a number532is = 30
D:\alagappauniv\JavaPrograms>
```

4. Write a Java program for swapping the two given numbers.

Program

```
import java.util.*;
class SwappingNumbers
{
```

NOTES

```
    public static void main(String args[])
    {
        int num1, num2, temp;
        Scanner scan= new Scanner(System.in);
        System.out.print("Please Enter No 1:=");
        num1=scan.nextInt();
        System.out.print("Please Enter No 2: =");
        num2=scan.nextInt();
        temp=num1;
        num1=num2;
        num2=temp;
        System.out.print("After Swapping...");
        System.out.println("First No: " + num1);
        System.out.println("Second No: " + num2);
    }
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac SwappingNumbers.java

G:\alagappauniv\JavaPrograms>java SwappingNumbers
Please Enter No 1:=34
Please Enter No 2: =55
After Swapping...First No: 55
Second No: 34

G:\alagappauniv\JavaPrograms>
```

5. Write a Java program for printing the Fibonacci series upto given number of terms.

Program

```
import java.util.*;
class FibonacciSeries
{
public static void main(String args[])
    {
        int a, b, temp, n;
        System.out.println("Program to display Fibonacci
Series...");
        Scanner scan= new Scanner(System.in);
        System.out.print("Please Enter No. =  ");
```

```
int t=scan.nextInt();
a=0;
b=1;
for(n=1;n<=t;n++)
{
System.out.println(a);
temp=a+b;
    a=b;
    b=temp;
}
            }          }
```

Output of the program:

```
D:\alagappauniv\JavaPrograms>java FibonacciSeries
Program to display Fibonacci Series...
Please Enter No. =  12
0
1
1
2
3
5
8
13
21
34
55
89
```

6. Write a Java program for generating a pattern using @ symbol.

Program

```
class PatternGenerator
{
public static void main(String[]args)
{
int row, countofSymbols;
 System.out.println("Pattern Generator Program");
for(row =1; row <=10; row++)
 {
for(countofSymbols=1;countofSymbols<=row;
countofSymbols++)
    {
System.out.print("@");
     }
System.out.println();// Go to next line
      }
    }
   }
```

NOTES

Output of the program:

```
Command Prompt

D:\alagappauniv\JavaPrograms>javac PatternGenerator.java

D:\alagappauniv\JavaPrograms>java PatternGenerator
Pattern Generator Program
@
@@
@@@
@@@@
@@@@@
@@@@@@
@@@@@@@
@@@@@@@@
@@@@@@@@@
@@@@@@@@@@

D:\alagappauniv\JavaPrograms>
```

7. Write a Java program for generating the prime number series.

Program

```
import java.util.*;
public class PrimeNumberSeries
{
public static void main(String args[])
{
int i,j,p=1;
Scanner sc =new Scanner(System.in);
System.out.print("Enter a number up to which you wish to
print prime numbers=");
int n=sc.nextInt();
System.out.print("The prime number series is as :");
for(i=2;i<=n;i++)
{
p=1;
for(j=2;j<i;j++)
{
if(i%j==0)
{
p=0;
}}
if(p==1)
System.out.print("\t"+j+",");
}
}
}
```

Output of the program:

```
Command Prompt

D:\alagappauniv\JavaPrograms>javac PrimeNumberSeries.java

D:\alagappauniv\JavaPrograms>java PrimeNumberSeries
Enter a number up to which you wish to print prime numbers=22
The prime number series is as : 2,      3,      5,      7,      11
,       13,     17,     19,
D:\alagappauniv\JavaPrograms>
```

8. Write a Java program for finding whether the given string is palindrome or not.

Program

```
// Class to find whether string is palindrome or not.
class PalindromeFinder
{
public static void main(String args[])
{
// Accepting the string at run time.
String s=args[0];
String s1=""; int l,j;
// Finding the length of the string.
l=s.length();
// Loop to find the reverse of the string.
for(j=l-1;j>=0;j–)
{ s1=s1+s.charAt(j);
}
// Condition to find whether two strings are equal // and
display the message.
if(s.equals(s1))
System.out.println("String "+s+" is palindrome");
else
System.out.println("String "+s+" is not palindrome");
}
}
```

Output of the program:

```
Command Prompt

D:\alagappauniv\JavaPrograms>javac PalindromeFinder.java

D:\alagappauniv\JavaPrograms>java PalindromeFinder alagappa
String alagappa is not palindrome

D:\alagappauniv\JavaPrograms>
```

9. Write a Java program for finding the quadratic roots and their types.

NOTES

Program

```
import java.lang.*;
import java.util.*;
class QuadraticRootFinder
{
public static void main(String args[])
{
int a,b,c,d;
double r1,r2;
      System.out.println("Program for finding the
Quadratic Roots ...");
      Scanner scan= new Scanner (System.in);
      System.out.print("Please Enter First Number: ");
      a=scan.nextInt();
      System.out.print("Please Enter Second Number: ");
      b=scan.nextInt();
      System.out.print("Please Enter Third Number: ");
      c=scan.nextInt();
d=b*b-4*a*c;
if(d==0)
{
r1=r2=-(float)b/(2*a);
System.out.println("the roots are real equal =" +r1 + "
and r2 = " +r2);
}
else if(d>0)
{
double t=Math.sqrt(d);
r1=(-b+t)/(2*a);

r2=(+b+t)/(2*a);
System.out.println("the roots are real and distict \n
r=" +r1 +" and r2=" +r2);
}
else if(d<0)
{
```

```
System.out.println("the roots are imaginary and there is
no real solution ");
}
}}
```

Output of the program:

```
Command Prompt

E:\alagappauniv\JavaPrograms>javac QuadraticRootFinder.java

E:\alagappauniv\JavaPrograms>java QuadraticRootFinder
Program for finding the Quadratic Roots ...
Please Enter First Number: 23
Please Enter Second Number: 90
Please Enter Third Number: 11
the roots are real and distict
 r=-0.1262986800131142 and r2=3.7867447982477556

E:\alagappauniv\JavaPrograms>java QuadraticRootFinder
Program for finding the Quadratic Roots ...
Please Enter First Number: 897
Please Enter Second Number: 877
Please Enter Third Number: 435
the roots are imaginary and there is no real solution

E:\alagappauniv\JavaPrograms>
```

NOTES

10. Write a Java program to print the product of two given matrices using arrays.

Program

```
import java.util.*;
import java.io.*;
import java.lang.*;
class MatrixMul
{
    public static void main(String[] args)
    {

       int i= 0,j=0,k=0,p,q,m,n;
       int a[][] = new int[10][10];
       int b[][] = new int[10][10],c[][] = new int[10][10];
       Scanner sc = new Scanner(System.in);
    System.out.print("Enter no of rows in the A matrix=
");
       p=sc.nextInt();
    System.out.print("Enter a no of columns in the A matrix
= ");
       q=sc.nextInt();
```

NOTES

```
    System.out.print("Enter no of rows in the B matrix=
");
       m = sc.nextInt();
    System.out.print("Enter no of columns in the B matrix=
");
       n = sc.nextInt();
                                    if(q==m)
                                    {
       System.out.print ("Enter A matrix values :");
       for(i=0;i<p;i++)
       {
            for(j=0; j<q ;j++)
                {
                    a[i][j] = sc.nextInt() ;
                }

       }

       System.out.println("\nMatrix A values are  ");

       for(i=0;i<p;i++)
       {
            for(j=0;j<q;j++)
                  {
          System.out.print(" "+a[i][j]) ;
                    }
          System.out.println();
              }
       System.out.println("Enter matrxi B values :");
       for(i=0;i<m;i++)
       {
              for(j=0;j<n;j++)
                      {
           b[i][j] = sc.nextInt() ;
          }
       }

                        System.out.println("\nMatrix B
values ");
       for(i=0;i<m;i++)
```

NOTES

```
        {
             for(j=0;j<n;j++)
                     {
           System.out.print(" "+b[i][j]) ;
           }
           System.out.println();
                          }
        System.out.println("\n C Matrix is C = A * B
Result");
        if(q == m)
        {
          for(i=0;i<p;i++)
                {
                   for(j=0;j<n;j++)
           {
              c[i][j]= 0 ;
             for(k=0;k<q;k++)
                {
                 c[i][j] = c[i][j] + (a[i][k] * b[k][j])
;
                 }
                                              }
              }
        }
        // for C matrix Printing

        for(i=0;i<p;i++)
        {
            for(j=0;j<n;j++)
                {
                   System.out.print(" "+c[i][j]) ;
                }
                                    System.out.println();
        }
                                }else
    {
         System.out.println("Matrix multiplication not
possible");
    }       }
}
```

NOTES

Output of the program:

```
C:\Windows\system32\cmd.exe

E:\alagappauniv\JavaPrograms>javac MatrixMul.java

E:\alagappauniv\JavaPrograms>java MatrixMul
Enter no of rows in the A matrix= 2
Enter a no of columns in the A matrix = 2
Enter no of rows in the B matrix= 2
Enter no of columns in the B matrix= 1
Enter A matrix values :23
33
45
56

Matrix A values are
 23 33
 45 56
Enter matrxi B values :
43
22

Matrix B values
 43
 22

 C Matrix is C = A * B Result
 1715
 3167

E:\alagappauniv\JavaPrograms>
```

```
E:\alagappauniv\JavaPrograms>java MatrixMul
Enter no of rows in the A matrix= 23
Enter a no of columns in the A matrix = 44
Enter no of rows in the B matrix= 22
Enter no of columns in the B matrix= 32
Matrix multiplication not possible

E:\alagappauniv\JavaPrograms>
```

11. Write a Java program for generating a pyramid of numbers.

Program

```
class NumberPyramid
{
public static void main(String[]args)
{
```

NOTES

```
int row, numberOfStars,n=0;
for(row =1; row <=5; row++)
 {
for(numberOfStars=1;
numberOfStars<=row;
numberOfStars++)
    {
System.out.print(numberOfStars);
     }
System.out.println();// Go to next line
      }
    }
   }
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac NumberPyramid.java

G:\alagappauniv\JavaPrograms>java NumberPyramid
1
12
123
1234
12345

G:\alagappauniv\JavaPrograms>
```

Try Yourself Questions

1. Write a Java program for finding the largest of five numbers.
2. Write a Java program for finding the factorial of any number.
3. Write a Java program for finding the product of three input digits.
4. Write a Java program for swapping the four input numbers.
5. Write a Java program for displaying the Fibonacci series up to *N*th Fibonacci numbers.
6. Write a Java program to generate random number.
7. Write a Java program for generating a pattern using * symbol.
8. Write a Java program for generating the series of 'Even' and 'Odd' numbers.
9. Write a Java program for finding whether the given string is palindrome or not.
10. Write a Java program to print the product of three given matrices using arrays.
11. Write a Java program for generating a pyramid pattern of numbers.

BLOCK II: OOP CONCEPTS

NOTES

This block contains the programs related to the following topics:

1. Function Overloading
2. Inheritance, Packages and Interfaces
3. Polymorphism

Function Overloading

Function overloading is used to reduce complexity and increase the efficiency of the program by involving more functions that are segregated and can be used to distinguish among each other with respect to their individual functionality. Overloaded functions are related to compile-time or static polymorphism. Function overloading in Java takes place when there are functions having the same name but have the different numbers of parameters passed to it which can be different in datatype like int, double, float and are used to return different values which are computed inside the respective overloaded method. There is also a concept of type conversion which is basically used in overloaded functions used to calculate the conversion of type in variables.

Advantages of Function Overloading

1. Function overloading works with the same name. So you do not have to create methods again which is already defined inside a respective function.
2. The functionality not only resolves the problem of conflicting naming but also improves the readability of the program.

1. Write a Java program for displaying the volume of cuboids when their dimensions are given.

Program

```
// program on function overloading
class CuboidEx
{
int width, breadth, height;
void getdata(int length)
    {
    width=breadth=height=length;
    }
    void getdata(int a,int b)
    {  width=a;
       breadth=height=b;
    }
```

```
void getdata(int x,int y, int z)
    {     width=x;
       breadth=y;
                 height=z;
    }
    int volume()
    {
System.out.println(" ");
    return width*breadth*height;
       }}
class MainCuboidEx
{   public static void main(String args[])
    {     int vol;
       CuboidEx myCuboidEx1=new CuboidEx();
                CuboidEx myCuboidEx2=new CuboidEx();
                CuboidEx myCuboidEx3=new CuboidEx();
       myCuboidEx1.getdata(120,220,330);
              vol=myCuboidEx1.volume();
            System.out.println("volume of my Cuboid with
3 parameters is =    " +vol);
       myCuboidEx2.getdata(10,20);
        vol=myCuboidEx2.volume();
           System.out.println("volume of my Cuboid with
2 parameters is =    " +vol);
                 myCuboidEx3.getdata(10);
        vol=myCuboidEx3.volume();
        System.out.println("Volume of my Cuboid with 1
parameter is =    " +vol);
    }
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac MainCuboidEx.java

G:\alagappauniv\JavaPrograms>java MainCuboidEx

volume of my Cuboid with 3 parameters is =    8712000

volume of my Cuboid with 2 parameters is =    4000

Volume of my Cuboid with 1 parameter is =    1000

G:\alagappauniv\JavaPrograms>
```

NOTES

2. Write a Java program for displaying the perimeter of rectangles when their dimensions are given.

Program

```
class RectangleExample
{
   int length;
   int width;
   int area()          // method for returning area of
rectangle
    {
     return length*width;
    }
   void perimeter()        // method for displaying the
perimeter of rectangle
    {
      int p=2*length+2*width;
       System.out.println("perimeter of Rectangle  is "
+p);
    }
        RectangleExample()
     {
         length=10;
         width=10;
      }
        RectangleExample(int a)
          {
              length=width=a;
            }
         RectangleExample(int a, int b)
          {
              length=a;
              width=b;
           }
}
  class MainForRectangleExample          //Main method
{
public static void main(String args[])
{
RectangleExample r1= new RectangleExample();
r1.length=5;
```

```
r1.width=5;
RectangleExample r2=new RectangleExample(10);
RectangleExample r3=new RectangleExample(20,10);
System.out.println(" ");
r1.perimeter();
r2.perimeter();
r3.perimeter();
float area1= r1.area();
float area2=r2.area();
System.out.println(" ");
System.out.println("area of Rectangle 1 and 2 are "
+area1+", " +area2);
System.out.println(" ");
System.out.println("area of Rectangle 3 is" +" "+r3.area());
}
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac MainForRectangleExample.java

G:\alagappauniv\JavaPrograms>java MainForRectangleExample

perimeter of Rectangle  is 20
perimeter of Rectangle  is 40
perimeter of Rectangle  is 60

area of Rectangle 1 and 2 are 25.0, 100.0

area of Rectangle 3 is 200

G:\alagappauniv\JavaPrograms>
```

3. Write a Java program for displaying back the already entered employee information.

Program

```
class Employee
{
    int id;
    String name;
    int age;
    Employee(int i,String n,int a)
   {
    id = i;
    name = n;
```

NOTES

```
        age=a;
        }

        Employee(Employee s)
{
        id = s.id;
        name =s.name;
        age=s.age;
        }
        void display()
{
System.out.println(id+" "+name+" "+age);
}
   public static void main(String args[])
{
                    Employee     s1     =     new
Employee(121,"AlagappaUniversityEmployee",39);
    Employee s2 = new Employee(s1); //copy contents from
s1 in to s2
    s1.display();
    s2.display();
   }
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac Employee.java

G:\alagappauniv\JavaPrograms>java Employee
121 AlagappaUniversityEmployee 39
121 AlagappaUniversityEmployee 39

G:\alagappauniv\JavaPrograms>
```

4. Write a Java program to illustrate the concept of operator precedence.

Program

```
classOperatorPrecedenceExample
  {
public static void main (String[] args)
      {
int result = 0;
         result = 15 + 21 * 3 - 10;
System.out.println("15 + 21 * 3 - 10= " +result +"\n");
```

```
        result = 35 + 44 /52 + 6;
System.out.println("35 + 44 /52 + 6= " +result+"\n");

        result = 32 + 63 / 21 * 23 - 221 + 23;
System.out.println("32 + 63 / 21 * 23 - 221 + 23= "
+result+"\n");

        result = 63 / 42 * 53 * 22 / 33;
System.out.println("63 / 42 * 53 * 22 / 33= " +result+"\n");

int x = 20;
        result = x++ + x++ * —x / x++ - —x + 3 >> 1 | 2;
System.out.println("result =" +result+"\n");

      }
  }
```

NOTES

Output of the program:

```
Command Prompt

C:\Users\Prat\Desktop\Alagappa University>javac OperatorPrecedenceExample.java

C:\Users\Prat\Desktop\Alagappa University>java OperatorPrecedenceExample
15 + 21 * 3 - 10= 68

35 + 44 /52 + 6= 41

32 + 63 / 21 * 23 - 221 + 23= -97

63 / 42 * 53 * 22 / 33= 35

result =11


C:\Users\Prat\Desktop\Alagappa University>
```

Inheritance

In Java, inheritance is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object-Oriented Programming system). In Java it helps you to create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Advantages of Inheritance

1. Inheritance is a process of defining a new class based on an existing class by extending its common data members and methods.
2. Inheritance allows to reuse of code, and it improves reusability in the Java application.

NOTES

3. Inheritance is a process for method overriding so that the runtime polymorphism can be achieved.
4. The biggest advantage of inheritance is that the code that is already present in base class need not be rewritten in the child class.

5. Write a Java program for demonstrating the concept of multilevel inheritance.

Program

```
class Name
 {
String name="Pratyash";
 int age=5;
}
class Marks extends Name
{
 int m1=80,m2=90,m3=30;
}
class Student extends Marks
{
int total;
void calc()
{
total=m1+m2+m3;
}
void show()
{
System.out.println("\n NAME: " +name+"\n AGE:"+age+"\n
MARK1="+m1+"\n   MARK2="  +m2+"\n   MARK3="+m3+"\n
TOTAL:"+total);
 }
}
class MultilevelInheritenceEx
 {
public static void main(String args[])
 {
Student ob=new Student();
ob.calc();
ob.show();
 }
 }
```

NOTES

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac MultilevelInheritenceEx.java

G:\alagappauniv\JavaPrograms>java MultilevelInheritenceEx

  NAME: Pratyash
  AGE:5
  MARK1=80
  MARK2=90
  MARK3=30
  TOTAL:200

G:\alagappauniv\JavaPrograms>
```

6. Write a Java program for illustrating the concept of abstract classes.

Program

```
abstract class Vehicle
{
public abstract void wheels();
public abstract void seating();
public abstract void brakes();
 }
class Car extends Vehicle
 {
public void wheels()
{
 System.out.println("\nCar Has Four Wheels");
 }
 public void seating()
 {
 System.out.println("Car Has Four Seating Capacity");
}
public void brakes()
{
System.out.println("Car Has Power Brakes\n");
 } }
class Bike extends Vehicle
{
 public void wheels()
{
```

NOTES

```
System.out.println("Bike Has Two Wheels");
 }
public void seating()
 {
System.out.println("Bike Has Two Seating Capacity");
}
public void brakes()
{
 System.out.println("Bike Has Disk Brakes");
 } }
class VehicleDemo
{
public static void main(String args[])
{
Vehicle v=new Car();
Vehicle v1=new Bike();
v.wheels();
v.seating();
v.brakes();
 v1.wheels();
v1.seating();
v1.brakes();
 } }
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac VehicleDemo.java

G:\alagappauniv\JavaPrograms>JAVA VehicleDemo

Car Has Four Wheels
Car Has Four Seating Capacity
Car Has Power Brakes

Bike Has Two Wheels
Bike Has Two Seating Capacity
Bike Has Disk Brakes

G:\alagappauniv\JavaPrograms>
```

7. Write a Java program for explaining the concept of `Super` keyword.

Program

```
class Super
{
```

```
static final int a=202;
static int b;
 int c=1030;
}
class SuperDemo extends Super
{
void display()
{
System.out.println("\nFinal Variable=" +super.a);
System.out.println("Super class Variable=" +super.c);
 System.out.println("Static Variable="+b);
 }
public static void main(String[] args)
{
SuperDemo s=new SuperDemo();
 s.display();
 } }
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac Super.java

G:\alagappauniv\JavaPrograms>java SuperDemo

Final Variable=202
Super class Variable=1030
Static Variable=0

G:\alagappauniv\JavaPrograms>
```

Packages

A package is a group of similar types of classes, interfaces and sub-packages. Package can be categorized into two types, i.e., built-in package and user-defined package. Keyword **import** is used to access a package from the library.

Advantages of a Package

1. Package is used to categorize the classes and interfaces so that they can be easily maintained.
2. Package provides access protection.
3. Package removes naming collision.

NOTES

8. Write a Java program to illustrate the concept of user defined packages.

Program

```
     //  save below file Sum.java
    package pack;
      public class Sum
           {
                public void getSum(int a,int b)
               {
                        System.out.println("\nSum Of Two
Numbers"+a+" and "+b+"="+(a+b));
              }
           }

        // save below file PackDemo.java
                       import pack.Sum;
class PackDemo
{
 public static void main(String[] args)
{
Sum s=new Sum();
s.getSum(10,20);
} }
```

Output of the program:

```
C:\WINDOWS\system32\cmd.exe
C:\javaprg\pack>javac -d . Sum.java
C:\javaprg\pack>javac PackDemo.java
C:\javaprg\pack>java PackDemo
Sum Of Two Numbers10 and 20=30
C:\javaprg\pack>_
```

9. Write a Java package program for class book and then import the data from the package and display the result.

Program

```
 //  Save below file Book.java
        package packagetest;
        public class Book
              {
              int book_no,book_id,book_pages;
             public Book(int a,int b, int c)
{
```

```
book_no=a;
 book_id=b;
book_pages=c;
}
                public void book_info()
{
System.out.println("The Book No"+book_no);
System.out.println("The Book Id "+book_id);
System.out.println("The Book Pages"+book_pages);
}
                    }
    //Save below file BookMain.java

         import packagetest.Book;
          class BookReader
                         {
                        public static void main(String[]
args)
                                {
                          Book b=new Book(12,15,86);
                         b.book_info();
                               }
                   }
```

Output of the program:

```
C:\WINDOWS\system32\cmd.exe
C:\javaprg\packagetest>javac BookReader.java

C:\javaprg\packagetest>java BookReader
The Book No12
The Book Id 15
The Book Pages86

C:\javaprg\packagetest>
```

10. Write a Java program for finding the cube of a number using package for various data types and then import it in another class and display the results.

Program

```
//1.program code to create package
package mathematics;
public class Mathmethods
{
```

NOTES

```
public static float Cube(float n)
{
return(n*n*n);
}
public static int Cube(int n)
{
return(n*n*n);
}
public static double Cube(double n)
{
return(n*n*n);
}
public static long Cube(long n)
{
return(n*n*n);
}
}
//2. Program code to import package
import mathematics.Mathmethods;
//import java.io.*;
import java.util.Scanner;
class Cube
{
public static void main(String S[])
{
//int a=20;
Scanner m=new Scanner(System.in);
System.out.println("the given number is ");
int a=m.nextInt();
Mathmethods mm = new Mathmethods();
int b = Mathmethods.Cube(a);
System.out.println("cube is " +b);
}
}
```

Output of the program:

javac –d . Mathmethods.java

javac Cube.java

java Cube

```
C:\WINDOWS\system32\cmd.exe

C:\javaprg\mathematics>javac Cube.java

C:\javaprg\mathematics>java Cube
the given number is
12
cube is 1728

C:\javaprg\mathematics>java Cube
the given number is
10
cube is 1000

C:\javaprg\mathematics>
```

Interfaces

Interface is a blueprint of a Class that can contain constants and abstract methods. They cannot be instantiated and can only be implemented by classes or extended by other interfaces. Syntax for defining interfaces is given below:

```
interface <interface_name>
{
    // declare constant fields
   // declare methods that abstract
   // by default.
}
```

Advantages of Interface

1. Interface can be used for the abstraction.
2. Interface supports the functionality of multiple inheritance.
3. Interface allows you to make your application loosely coupled.

11. Write a Java program for demonstrating the use of interfaces.

Program

```
import java.lang.*;
 import java.io.*;
interface Exam
{
void percent_cal();
}
class Student
{
 String name;
int roll_no,mark1,mark2;
Student(String n, int r, int m1, int m2)
 {
```

NOTES

```
name=n;
roll_no=r;
mark1=m1;
mark2=m2;
}
void display()
 {
System.out.println ("\nName of Student: "+name);
System.out.println ("Roll No. of Student: "+roll_no);
System.out.println ("Marks of Subject 1: "+mark1);
System.out.println ("Marks of Subject 2: "+mark2);
 } }
class Result extends Student implements Exam
{
Result(String n, int r, int m1, int m2)
{
super(n,r,m1,m2);
}
public void percent_cal()
{
int total=(mark1+mark2);
 float percent=total*100/200;
System.out.println ("Percentage: "+percent+"%");
 }
void display()
{
super.display();
 } }
 class MultipleInheritenceDemo
{
public static void main(String args[])
{
 Result R = new Result("Sunayna Alangary",62,83,84);
 R.display();
 R.percent_cal();
 } }
```

NOTES

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac MultipleInheritenceDemo.java

G:\alagappauniv\JavaPrograms>java MultipleInheritenceDemo

Name of Student: Sunayna Alangary
Roll No. of Student: 62
Marks of Subject 1: 83
Marks of Subject 2: 84
Percentage: 83.0%

G:\alagappauniv\JavaPrograms>
```

12. Write a program on the concept of polymorphism.

Program

```
class Person {
    String name;

    void know() {
System.out.println("My name is "+ name+ " and I know many
things");
     }
    void expertise() {
System.out.println("My name is "+ name+ " and I am expert
in computer science");
     }
 }

 class Teacher extends Person {
    String subject;

    void teach() {
System.out.println("I teach  "+subject+"to my students");
     }
    public static void main(String [] args) {
     Teacher teacher = new Teacher(); //new instance of
the Teacher class
      teacher.name = "MOHAN RAO";
teacher.subject = "Java";
teacher.know();
teacher.expertise();
teacher.teach();
```

```
      Person person = teacher; // polymorphism here as
the teacher object now takes the form of person object
person.know ();
person.expertise();
      //person.teach(); // Can't call teach method as
it's not in Person class , So it has been made comment
    }
 }
```

NOTES

Output of the program:

```
Command Prompt

C:\Users\Prat\Desktop\Alagappa University>javac Teacher.java
Teacher.java:29: error: cannot find symbol
      person.teach();
            ^
  symbol:   method teach()
  location: variable person of type Person
1 error

C:\Users\Prat\Desktop\Alagappa University>javac Teacher.java

C:\Users\Prat\Desktop\Alagappa University>java Teacher
My name is MOHAN RAO and I know many things
My name is MOHAN RAO and I am expert in computer science
I teach  Javato my students
My name is MOHAN RAO and I know many things
My name is MOHAN RAO and I am expert in computer science

C:\Users\Prat\Desktop\Alagappa University>
```

13. Write a program to demonstrate compile time polymorphism in Java.

Program

```
 class CompileTimePolymorphismEx {
    void multiply(int a, int b) {
int product = a * b;
System.out.println("The Product of two numbers =
"+product);
     }
     void multiply(int a, int b, int c) {
int product = a * b * c;
System.out.println("The Product of three numbers =
"+product);
     }
       void multiply(int a, int b, intc,int d) {
int product = a * b * c * d;
System.out.println("The Product of four numbers =
"+product);
     }
    public static void main(String [] args) {
```

NOTES

```
CompileTimePolymorphismExobj = new
CompileTimePolymorphismEx();
obj.multiply(320,230);
      obj.multiply(220,330,404);
         obj.multiply(3,5,6,7);
    }
 }
```

Output of the program:

```
C:\Users\Prat\Desktop\Alagappa University>javac CompileTimePolymorphismEx.java

C:\Users\Prat\Desktop\Alagappa University>java CompileTimePolymorphismEx
The Product of two numbers = 73600
The Product of three numbers = 29330400
The Product of four numbers = 630

C:\Users\Prat\Desktop\Alagappa University>
```

14. Write a program to perform runtime polymorphism in Java.

Program

```
 class Bird {
   void fly()
   {
System.out.println("Ia m a bird  and I can fly ");
   }
 }

 class Eagle extends Bird {
   void fly()
   {
System.out.println("I am an Eagle and I can  fly very
high!");
   }
   public static void main(String [] args) {
     Bird bird = new Bird();
bird.fly(); //basic feature of bird is to fly() called
from parent Bird class
     Eagle eagle= new Eagle();
eagle.fly();    //eagle is overriding the super class
bird fly()
    }
 }
```

NOTES

Output of the program:

```
Command Prompt

C:\Users\Prat\Desktop\Alagappa University>javac Eagle.java

C:\Users\Prat\Desktop\Alagappa University>java Eagle
Ia m a bird  and I can fly
I am an Eagle and I can  fly very high!

C:\Users\Prat\Desktop\Alagappa University>
```

Try Yourself Questions

1. Write a Java program for displaying the volume of a rectangle when their dimensions are given.
2. Write a Java program to find area of square, cuboid, rectangle and circle using method overloading.
3. Write a Java program for displaying the perimeter of 5 rectangles when their dimensions are given.
4. Write a Java program for displaying back the student information already entered in student table.
5. Write a Java program to illustrate the instance of operator.
6. Write a Java program to illustrate max of three numbers using ternary operator.
7. Write a Java program to illustrate the concept of inheritance.
8. Write a Java program to illustrate the concept of multilevel inheritance.
9. Write a Java program to illustrate the concept of hierarchical inheritance.
10. Write a Java program using method overriding to demonstrate why we need to call the method of parent class with child class object.
11. Write a Java program for explaining the concept of `Super` keyword.
12. Write a Java program to illustrate the concept of user defined packages.
13. Write a Java program to demonstrate accessing of members when corresponding classes are imported and not imported.
14. Write a Java program for finding the square of a number using package for various data types and then import it in another class and display the results.
15. Write a Java program for providing the implementation of Bank interface.
16. Write a Java program to perform runtime polymorphism. Consider a scenario where Bank is a class that provides a method to get the rate of interest. However, the rate of interest may differ according to banks. For example, SBI, ICICI, and AXIS banks are providing 8.4%, 7.3%, and 9.7% rate of interest.
17. Write a Java program to demonstrate compile time polymorphism.

BLOCK III: THREAD AND VIRTUAL FUNCTION

This block contains the programs related to the following topics:

1. Threads
2. Virtual Functions

Threads are sometimes called as lightweight processes. It allows performing various tasks in the background at the same time without interrupting the main program. Following are the two ways of creating threads:

- Extending the Thread Class
- Overriding the **`run()`** Method

Virtual function is a function or method which is used in an inherited class to override the behaviour of the function having the same definition to achieve the polymorphism. In Java, **`virtual`** keyword is not used to define the virtual functions.

1. Write a Java program for demonstrating the threading concept.

Program

```
import java.io.*;
 class A extends Thread
{
public void run()
{
System.out.println("Start A");
 for(int i=1; i<=5; i++) System.out.println("Thread A i
:"+i);
System.out.println("Exit A");
}
}
class B extends Thread
{
public void run()
{
System.out.println("Start B");
for(int j=1; j<=5; j++) System.out.println("Thread B j
:"+j);
System.out.println("Exit B");
 }
}
class C extends Thread
{
```

NOTES

```
public void run()
{
System.out.println("Start C");
 for(int k=1; k<=5; k++) System.out.println("Thread C k
:"+k);
System.out.println("Exit C");
 }
 }
class D extends Thread
{
 public void run()
{
System.out.println("Start D");
 for(int l=1; l<=5; l++)
System.out.println("Thread A l :"+l);
System.out.println("Exit D");
} }
class threadtest
{
public static void main(String S[])throws IOException
{
new A().start();
 new B().start();
new C().start();
new D().start();
}}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac threadtest.java

G:\alagappauniv\JavaPrograms>java threadtest
Start A
Thread A i :1
Thread A i :2
Thread A i :3
Thread A i :4
Thread A i :5
Exit A
Start B
Thread B j :1
Thread B j :2
Thread B j :3
Thread B j :4
Thread B j :5
Exit B
Start C
Thread C k :1
Thread C k :2
Thread C k :3
Thread C k :4
Thread C k :5
Start D
Exit C
Thread A l :1
Thread A l :2
Thread A l :3
Thread A l :4
Thread A l :5
Exit D

G:\alagappauniv\JavaPrograms>
```

NOTES

2. Write a Java program for demonstrating the concept of multithreading.

Program

```
public class ThreadDemo2 extends Thread
{
 public void run()
{
for(int i=0;i<20;i++)
 {
 System.out.println(getName()+":"+i);
 }
 }
 public static void main(String[] args)
 {
 System.out.println("main started");
ThreadDemo2 td=new ThreadDemo2();
ThreadDemo2 td1=new ThreadDemo2();
td.setName("Thread1");
 td1.setName("Thread2");
td.start();
 td1.start();
 td.yield();
 System.out.println("Main Exited");
 }
 }
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac ThreadDemo2.java

G:\alagappauniv\JavaPrograms>java ThreadDemo2
main started
Main Exited
Thread1:0
Thread2:0
Thread1:1
Thread2:1
Thread1:2
Thread1:3
Thread1:4
Thread1:5
Thread1:6
Thread1:7
Thread2:2
Thread1:8
Thread2:3
Thread1:9
Thread1:10
```

NOTES

3. Write a Java program for understanding the concept of daemon threads.

Program

```
class DaemonThread extends Thread
{
 int i;
 public void run()
  {
 for(i=0;i<5;i++);
{
 System.out.println(this.getName()+" "+i);
}
}
 public static void main(String[] args)
  {
DaemonThread d=new DaemonThread();
DaemonThread d1=new DaemonThread();
d.setName("\nDAEMON THREAD");
d1.setName("NORMAL THREAD");
d.setPriority(Thread.MIN_PRIORITY);
d.start();
d1.start();
}
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac DaemonThread.java

G:\alagappauniv\JavaPrograms>java DaemonThread
NORMAL THREAD 5

DAEMON THREAD 5

G:\alagappauniv\JavaPrograms>
```

4. Write a program using virtual methods/functions in Java.

Program

```
class Faculties {

    //virtual method
    void teach() {
```

```
      System.out.println("CAN BE ANY FACULTY!");
   }
}

public class CSFaculty extends Faculties {
   void teach() {
      System.out.println("Computer Science Faculty
ONLY");
   }

   public static void main (String args[]) {
      Faculties newfaculty=new CSFaculty();
      newfaculty.teach();
   }
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac DaemonThread.java

G:\alagappauniv\JavaPrograms>java DaemonThread
NORMAL THREAD 5

DAEMON THREAD 5

G:\alagappauniv\JavaPrograms>
```

Try Yourself Questions

1. Write a Java program for demonstrating the threading concept. Demonstrate that how Java creates a new thread and starts running it successfully.
2. Write a Java program for demonstrate how to extend the thread.
3. Write a Java program to demonstrate the usage of exception in `Daemon()` thread.
4. Write a Java program for demonstrating the concept of multithreading.
5. Write a Java program to demonstrate the usage of `setDaemon()` and is`Daemon()` method.
6. Write a program using virtual methods in Java.
7. Write a program using virtual functions in Java.

BLOCK IV: I/O AND EXCEPTION HANDLING

NOTES

This block contains the program related to the following topics:

1. String and Exception Handling
2. I/O and Stream Handling

String and Exception Handling

The 'Exception Handling' in Java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained. In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

Basically, an exception (or exceptional event) is a problem that arises during the execution of a program. When an 'Exception' occurs in the program then the normal flow of the program is disrupted and the Program/Application terminates abnormally, which is not recommended and therefore these exceptions must be checked and controlled.

An exception can occur for many different reasons, such as a user has entered an invalid data, a file that needs to be opened cannot be found, a network connection has been lost in the middle of communications or the JVM has run out of memory, etc. Some of these exceptions are caused by user error, others by programmer error, and others by physical resources that have failed in some manner.

Advantage of Exception Handling

The core advantage of exception handling is to maintain the normal flow of the application. An exception normally disrupts the normal flow of the application that is why exception handling is used.

I/O and Stream Handling

The **java.io package** contains nearly every class essential to perform an Input and Output (I/O) in Java. All these streams represent an input source and an output destination. In Java, streams are the sequence of data that are read from the source and written to the destination. An input stream is used to read data from the source while an output stream is used to write data to the destination. Java byte streams are used to perform input and output of 8-bit bytes.

1. Write a Java program for sorting the strings.

Program

```
import java.lang.*;
class StringSorter
{
static String name[]={"University" , "Alagappa"};
```

NOTES

```
public static void main(String args[])
{
int size=name.length;
String temp=null;
for(int i=0;i<size;i++)
{
for(int j=i+1;j<size;j++)
{
if ( name[j].compareTo(name[i])<0)
{
temp=name[i];
name[i]=name[j];
name[j]=temp;
}
}
}
System.out.println("sorted names are  ");
for(int i=0;i<size;i++)
{
System.out.println(name[i]);
}
}
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac StringSorter.java

G:\alagappauniv\JavaPrograms>java StringSorter
sorted names are
Alagappa
University

G:\alagappauniv\JavaPrograms>
```

2. Write a Java program for sorting names using array.

Program

```
// String array sorting
class Sort
{
    static String s;
public static void main(String aa[])
```

NOTES

```
{
          String   ar[]   =   new
String[]{"Sumit","Kapil","Amit","Imraan","Raja",
"Bhima","Hariom","Raka","Ankur","Pintu"};
for(inti=0;i<ar.length-1;i++)
    {
for(int j=i+1;j<ar.length;j++)
      {
if(ar[i].compareTo(ar[j])>0)
    {
           s = ar[i];
ar[i] = ar[j];
ar[j] = s;
        }
      }
    }
for(inti=0;i<ar.length;i++)
System.out.println(ar[i]);
  }
}
```

Output of the program:

```
C:\Users\Prat\Desktop\java\DG>Sort.java

C:\Users\Prat\Desktop\java\DG>javac Sort.java

C:\Users\Prat\Desktop\java\DG>java Sort
Amit
Ankur
Bhima
Hariom
Imraan
Kapil
Pintu
Raja
Raka
Sumit

C:\Users\Prat\Desktop\java\DG>
```

3. Write a Java program for iterating a list using collection.

Program

```
importjava.util.*;
public class ListIteratorDemo {
public static void main(String[] args) {
List<String> l1=new ArrayList<String>();
l1.add("apple");
```

```
l1.add(“nepal”);
l1.add(“calender”);
l1.add(“hospital”);
l1.add(“Mapple”);
l1.add(“Temple”);
l1.add(“Cripple”);
l1.add(“Alagappa”);
System.out.println(“list = “+l1);
Iterator<String> it=l1.iterator();
while (it.hasNext()) {
    System.out.println(it.next()+”\t”);
}
ListIterator<String> li=l1.listIterator(1);
while (li.hasNext()) {
    System.out.print(li.next()+”\t”);
}
}
}
```

Output of the program:

```
Command Prompt

C:\Users\Prat\Desktop\java\DG>javac ListIteratorDemo.java

C:\Users\Prat\Desktop\java\DG>java ListIteratorDemo
list = [apple, nepal, calender, hospital, Mapple, Temple, Cripple, Alagappa]
apple
nepal
calender
hospital
Mapple
Temple
Cripple
Alagappa
nepal   calender        hospital        Mapple  Temple  Cripple Alagappa
C:\Users\Prat\Desktop\java\DG>
```

4. Write a Java program for demonstrating the use of `StringTokenizer` class.

Program

```
import java.util.StringTokenizer;
import java.io.*;
 class StringTockenDemo
{
 public static void main(String args[]) throws IOException
{
BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
```

NOTES

```
System.out.println("Enter All Digits with Spaces");
String str=br.readLine();
 StringTokenizer st=new StringTokenizer(str);
int a,prod=1;
 String s;
 while(st.hasMoreTokens())
{
 s=st.nextToken();
a=Integer.parseInt(s);
 prod=prod*a;
 }
 System.out.println("Product of The given Integers"+prod);
}
 }
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac StringTockenDemo.java

G:\alagappauniv\JavaPrograms>java StringTockenDemo
Enter All Digits with Spaces
5 6 56
Product of The given Integers1680

G:\alagappauniv\JavaPrograms>java StringTockenDemo
Enter All Digits with Spaces
2 3 4
Product of The given Integers24

G:\alagappauniv\JavaPrograms>
```

5. Write a Java program for demonstrating the exception handling mechanism.

Program

```
class ExcHandling
{
public static void main(String ar[])
{
int a[]={2,10};
int b=5;
try
{
int x=a[2]/(b-a[0]);
}
catch(ArithmeticException e)
```

```
{
System.out.println("div by zero");
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("Array out of bound Error");
}
int y=a[1]/a[0];
System.out.println("y value is " +y);
}
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac ExcHandling.java

G:\alagappauniv\JavaPrograms>java ExcHandling
Array out of bound Error
y value is 5

G:\alagappauniv\JavaPrograms>
```

6. Write a Java program for demonstrating the divide by zero exception handling.

Program

```
import java.lang.*;
class Error2
{
public static void main(String args[])
{
int a=10;
int b=5;
int c=5;
int x,y;
try
{
 x=a/(b-c);
}
catch (ArithmeticException e)
{
System.out.println("division by zero");
}
```

NOTES

```
 y=a/(b+c);
System.out.println("y=" +y);
}
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac Error2.java

G:\alagappauniv\JavaPrograms>java Error2
division by zero
y=1

G:\alagappauniv\JavaPrograms>
```

7. Write a Java program for explaining the concept of exception handling using the insufficient balance exception while displaying the Bank balance.

Program

```
class InsufficientBalanceExceptionGen extends Exception
{
InsufficientBalanceExceptionGen(String s)
{
super(s);
}
}
class MainBalanceCheck
{
public static void main(String s[])
{
int bal=5000;
try
{
int withdraw=Integer.parseInt(s[0]);
    if(bal<withdraw)
    {
    InsufficientBalanceExceptionGen  i=  new
InsufficientBalanceExceptionGen("No such bance exist");
    throw i;
    }
    else
```

NOTES

```
    {
    System.out.println("Current balance is "+bal);
    bal=bal-withdraw;
    System.out.println("The balance after withdrawal is
"+bal);
    }
}
catch(InsufficientBalanceExceptionGen i)
{
System.out.println(i);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output of the program:

```
Command Prompt

G:\alagappauniv\JavaPrograms>javac MainBalanceCheck.java

G:\alagappauniv\JavaPrograms>java MainBalanceCheck 500
Current balance is 5000
The balance after withdrawal is 4500

G:\alagappauniv\JavaPrograms>java MainBalanceCheck 50000
InsufficientBalanceExceptionGen: No such bance exist

G:\alagappauniv\JavaPrograms>
```

File I/O and Streams

A file is a collection of related records placed in a particular area on disk. Storing and managing data using files is known as file processing which includes creating, updating files and manipulation of data. Reading and writing of data in a file can be done at the level of bytes or characters or fields. Java byte streams are typically used for performing specific input and output of 8-bit bytes. Though there are many classes related to byte streams but the most frequently used classes are, **`FileInputStream`** and **`FileOutputStream`**.

Input refers to the flow of data into a program and output means the flow of data out of a program. A stream in Java is a path along which data flows. The **java.io package** contains a large number of stream classes that provide capabilities for processing all type of data.

NOTES

8. Write a program for implementing a `FileInputStream` in Java.

Program

```
// FileInputStream example
import java.io.*;
class F1
{
public static void main(String aa[])
    {
int x;
try
      {
FileInputStream r = new FileInputStream("C://test1//
B.txt");// there should be a file named B.txt
while((x=r.read())!=-1) // while not eof
         {
    System.out.print((char)x);
Thread.sleep(1000);
     }
}catch(Exception ex){}
    }
}
```

Output of the program:



9. Write a Java program to demonstrate the use of `FileOutputStream` and `PrintStream` classes.

NOTES

Program

```
import java.io.*;

class FileOutput
{
public static void main(String args[])
{
FileOutputStream out; // declare a file output object
PrintStream p; // declare a print stream object

try
{
// Create a new file output stream connected to "myfile.txt"
out = new FileOutputStream("myfile.txt");

// Connect print stream to the output stream
p = new PrintStream( out );
p.println ("This is written to a file myFile.txt");
p.close();
}
catch (Exception e)
{
System.err.println ("Error writing to the file
myFile.txt");
}
}
}
```

Output of the program:



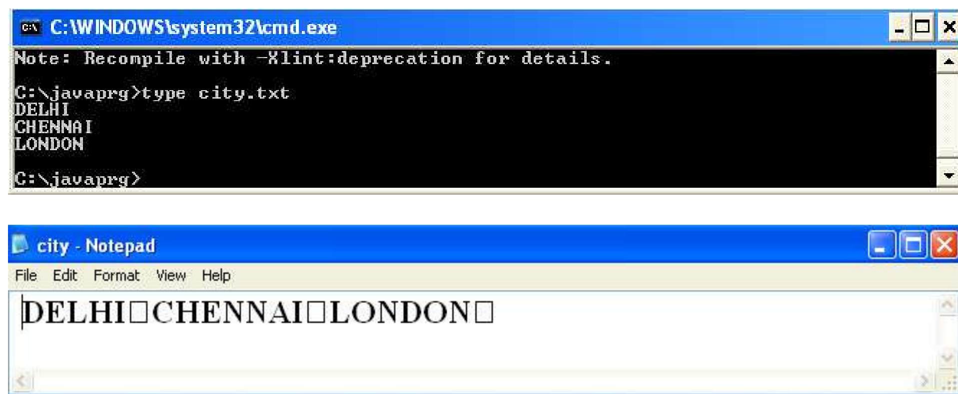
NOTES

10. Write a Java program for writing bytes to a file.

Program

```
import java.io.*;
class WriteBytes
{
public static void main(String args[])
{
// declare and initialize a byte array
byte cities[]={'D','E','L','H','I','\n','C','H','E','N','N','A','I','\n','L','O','N','D','O','N','\n'};
//create an output file stream
FileOutputStream outfile=null;
try
{
//connect the outfile stream to city.txt
outfile=new FileOutputStream("city.txt");
//write data to the stream
outfile.write (cities);
outfile.close( );
}
catch(IOException ioe)
{
System.out.println(ioe);
System.exit(-1);
}
}
}
```

Output of the program:



11. Write a Java program to copy one file to another.

Program

```
//copy bytes from one to another
import java.io.*;
class CopyBytes
{
public static void main(String args[])
{
//declare input and output file streams
FileInputStream infile=null;
FileOutputStream outfile=null;
//declare a varible to hold a byte
byte byteRead;
try
{
//Connect infile to in.dat
infile = new FileInputStream("in.dat");
//Connect outfile to in.dat
outfile=new FileOutputStream("out.dat");
//Reading bytes from in.dat and writing to out.dat
do
{
byteRead=(byte)infile.read( );
outfile.write(byteRead);
}
while(byteRead !=-1);
}
catch(FileNotFoundException e)
{
System.out.println("File not fopund");
}
catch(IOException e)
{
System.out.println(e.getMessage( ));
}
finally     //close files
{
try
{
infile.close( );
```

NOTES

NOTES

```
outfile.close( );
}
catch(IOException e)
{    }
}
}
}
```

Output of the program:

```
C:\WINDOWS\system32\cmd.exe
C:\javaprg>javac CopyBytes.java

C:\javaprg>java CopyBytes

C:\javaprg>type out.dat
java programming for internet
java script for web page development
perl for server side scripting
C:\javaprg>
```

Try Yourself Questions

1. Write a Java program for sorting the strings.
2. Write a Java Program to count the number of words in a string.
3. Write a Java program for sorting names using array.
4. Write a Java Program to find whether a string or number is palindrome or not.
5. Write a Java Program to find the duplicate characters in a string.
6. Write a Java program to demonstrate why collection framework is needed.
7. Write a Java program for demonstrating the working of `StringTokenizer` class.
8. Write a Java program for demonstrating the exception handling mechanism using the Java `throws` keyword.
9. Write a program for implementing a `FileInputStream` in Java.
10. Write a Java program to demonstrate the use of `FileOutputStream` and `PrintStream` classes.
11. Write a Java program for writing bytes to a file.
12. Write a Java program to convert a string into `byte[]` and write to a file.
13. Write a Java program to copy one file to another.
14. Write a Java program for explaining the concept of exception handling using the insufficient balance exception while displaying the Bank balance. Create an Account class, which represents a Bank account where you can deposit and withdraw money, and can check the Amount when you have to withdraw money weather it will exceed your bank balance. Using user defined exception create a custom exception called InSufficient Fund Exception to handle this scenario.

BLOCK V: NETWORK PROGRAMMING

This block contains the programs related to Applet, Swing and Networking.

Abstract Window Toolkit (AWT)

Abstract Window Toolkit (AWT) contains numerous classes and methods that allow you to create and manage windows. AWT is a platform-dependent API (Application Programming Interface) to develop GUI (Graphical User Interface) or Window-based applications in Java. A common use of the AWT is in Applets, it is also used to create stand-alone windows that run in a GUI environment, such as Windows.

Java Networking

Java networking is a concept of connecting two or more computing devices together so that the resources can be shared. Java socket programming provides facility to share data between different computing devices.

Advantage of Java Networking

1. Java networking helps in sharing resources.
2. Java networking centralizes the software management.

Applet

In Java, **JApplet** is a class that enables Applets to use Swing components. **JApplet** is a subclass of **java.applet.Applet** or any Applet that contains Swing components must be implemented with a subclass of **JApplet**.

Swing

Swing is a Java GUI that is used to create various applications. Swing has components which are platform-independent. It enables the user to create buttons and scroll bars. Swing includes packages for creating desktop applications in Java. Swing components are written in Java language and is a part of Java Foundation Classes (JFC).

Applet Programs

1. Write a Java Applet program demonstrating the ball bouncing animation.

Program

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
/*<applet code= "Bounce.class" height=900 width=900></
applet>*/
```

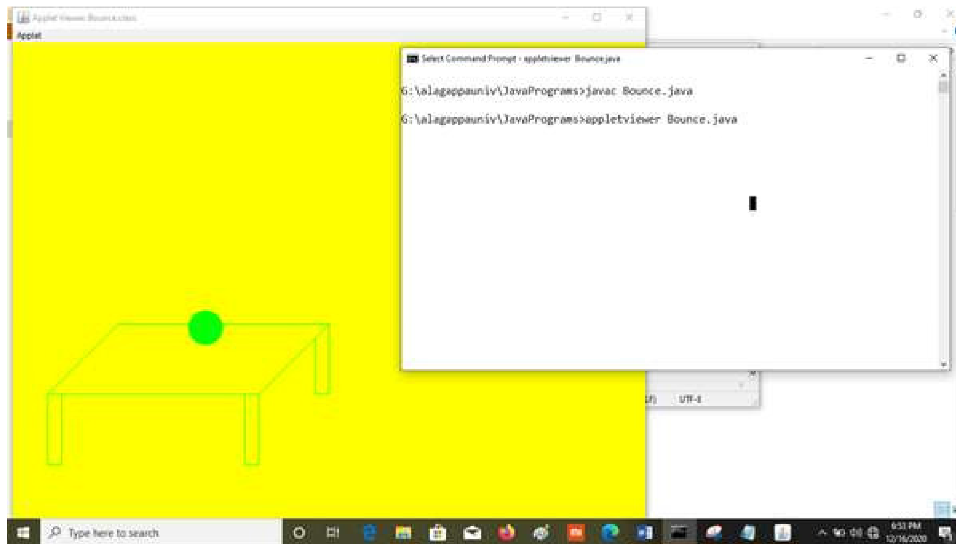
NOTES

```
 class Ball
{
int x,y,radius,dx,dy;
Color BallColor;
 public Ball(int x,int y,int radius,int dx,int dy,Color
bColor)
 {
this.x=x;
this.y=y;
this.radius=radius;
this.dx=dx;
this.dy=dy;
BallColor=bColor;
} }
 public class Bounce extends Applet implements Runnable
{
Ball redBall;
 public void init()
 {
redBall=new Ball(250,80,50,2,4,Color.green);
Thread t=new Thread(this);
 t.start();
}
 public void paint(Graphics g)
{
g.setColor(redBall.BallColor);
 setBackground(Color.yellow);
//g.setcolor(redBall.BallColor);
g.fillOval(redBall.x,              redBall.y,
redBall.radius,redBall.radius);
g.drawLine(150,400,50,500);
g.drawLine(150,400,450,400);
g.drawLine(50,500,350,500);
g.drawLine(450,400,350,500);
g.drawRect(50,500,20,100);
 g.drawRect(330,500,20,100);
g.drawLine(450,400,450,500);
g.drawLine(430,500,450,500);
g.drawLine(430,500,430,420);
}
public void run()
```

NOTES

```
{ while(true)
 {
try
 {
displacementOperation(redBall);
Thread.sleep(20);
repaint();
}
catch(Exception e)
{
} } }
public void displacementOperation(Ball ball)
{
if(ball.y >= 400 || ball.y <= 0)
{
ball.dy=-ball.dy; } ball.y=ball.y+ball.dy;
 }
}
```

Output of the program:



2. Write a Java Applet program for displaying a banner.

Program

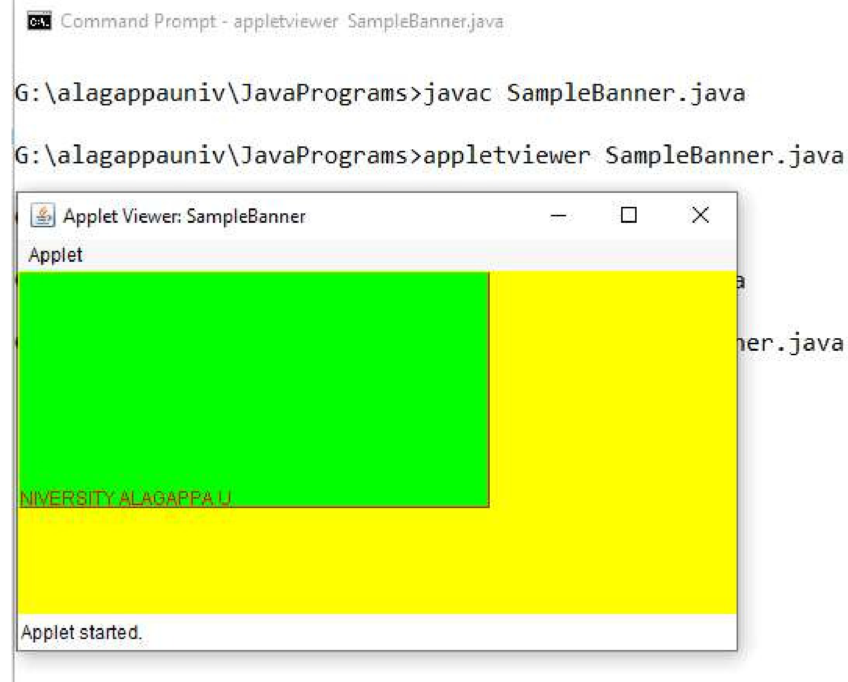
```
import java.awt.*;
import  java.applet.*;
 /*<HTML><BODY><APPLET CODE = “SampleBanner” WIDTH = “460”
```

NOTES

```
HEIGHT = "220"></APPLET></BODY></HTML> */
public class SampleBanner extends Applet implements
Runnable
{
 String str = "ALAGAPPA UNIVERSITY ";
Thread t ;
boolean b;
public void init()
{
setBackground(Color.YELLOW);
 setForeground(Color.RED);
}
public void start()
{
t = new Thread(this);
b = false; t.start();
}
public void run ()
{
char ch;
 for( ; ; )
{
try
 {
repaint();
Thread.sleep(250);
ch = str.charAt(0);
 str = str.substring(1, str.length());
 str = str + ch;
}
catch(InterruptedException e)
{}
}
}
 public void paint(Graphics g)
{
g.drawRect(1,1,300,150);
 g.setColor(Color.green);
g.fillRect (1,1,300,150);
g.setColor(Color.red);
```

```
 g.drawString(str, 1, 150);
} }
```

Output of the program:



NOTES

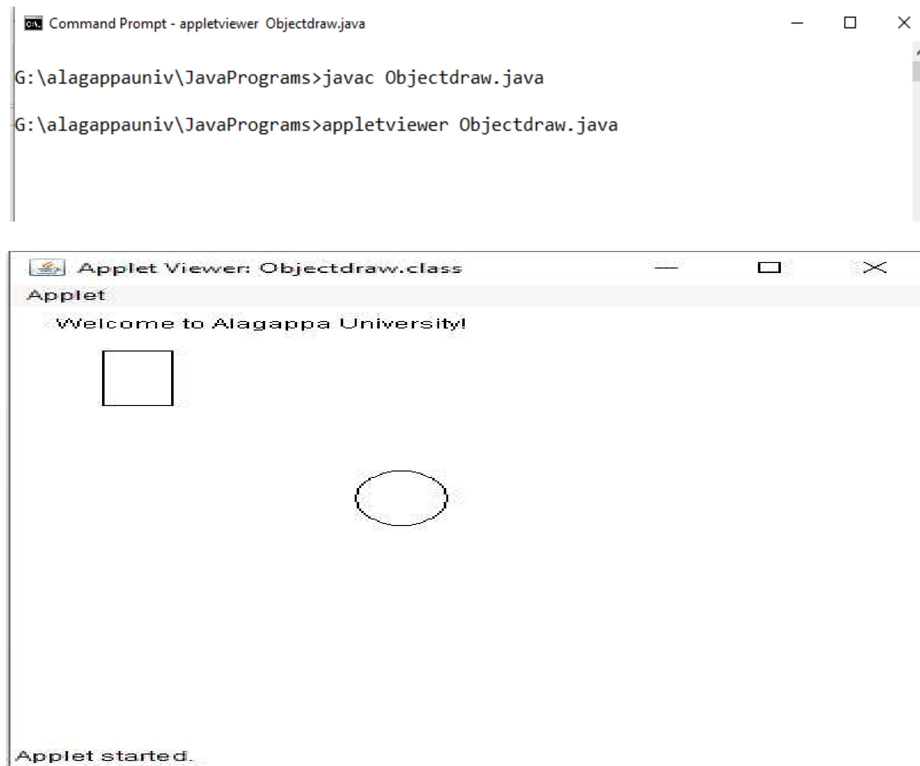
3. Write a Java Applet program for drawing objects and displaying strings.

Program

```
import java.awt.*;
import java.applet.*;
/*<applet code= "Objectdraw.class" height=400 width=400></
applet>*/
public class Objectdraw extends Applet
{
 public void paint(Graphics g)
{
g.drawString("Welcome to Alagappa University!",20,20);
 g.drawRect(40,40,30,50);
g.drawOval(150,150,40,50);
} }
```

NOTES

Output of the program:



4. Write a Java program that prints a message by clicking on the button using AWT Events and Applets.

Program

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class EventApplet extends Applet implements ActionListener
{
Button b;
TextField tf;
 public void init()
{
tf=new TextField();
tf.setBounds(30,40,150,20);

b=new Button("Click");
b.setBounds(80,150,60,50);

add(b);add(tf);
```

NOTES

```
b.addActionListener(this);

setLayout(null);
}

 public void actionPerformed(ActionEvent e)
{
  tf.setText("Welcome");
 }
}
```

In the above code, we have created all the controls in init() method because it is invoked only once.

```
myapplet.html
<html>
<body>
<applet code="EventApplet.class" width="300" height="300">
</applet>
</body>
</html>
```

Output of the program:



5. Write a Java Applet program for handling Mouse Events.

Program

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
/*<applet code="MouseDemo" width=300 height=300>
</applet>*/
public class MouseDemo extends Applet implements
```

NOTES

```
MouseListener,MouseMotionListener
{
int mx=0;
int my=0;
String msg="";
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent me)
{
mx=20;
my=40;
msg="Mouse Clicked";
repaint();
}
public void mousePressed(MouseEvent me)
{
mx=30;
my=60;
msg="Mouse Pressed";
repaint();
}
public void mouseReleased(MouseEvent me)
{
mx=30;
my=60;
msg="Mouse Released";
repaint();
}
public void mouseEntered(MouseEvent me)
{
mx=40;
my=80;
msg="Mouse Entered";
repaint();
}
public void mouseExited(MouseEvent me)
{
```

NOTES

```
mx=40;
my=80;
msg="Mouse Exited";
repaint();
}
public void mouseDragged(MouseEvent me)
{
mx=me.getX();
my=me.getY();
showStatus("Currently mouse dragged"+mx+" "+my);
repaint(); }
public void mouseMoved(MouseEvent me)
{
mx=me.getX();
my=me.getY();
showStatus("Currently mouse is at"+mx+" "+my);
repaint();
}
public void paint(Graphics g)
{
g.drawString("Handling Mouse Events",30,20);
g.drawString(msg,60,40);
}
}
```

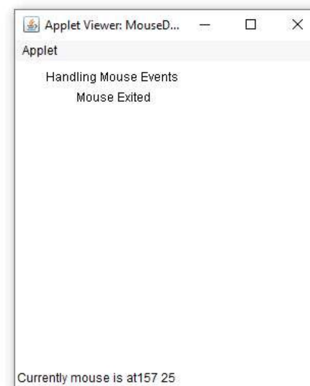
Output of the program:

```
Command Prompt - appletviewer MouseDemo.java

G:\alagappauniv\JavaPrograms>exam.java

G:\alagappauniv\JavaPrograms>javac MouseDemo.java

G:\alagappauniv\JavaPrograms>appletviewer MouseDemo.java
```



NOTES

6. Write a Java program for creating a notepad like application.

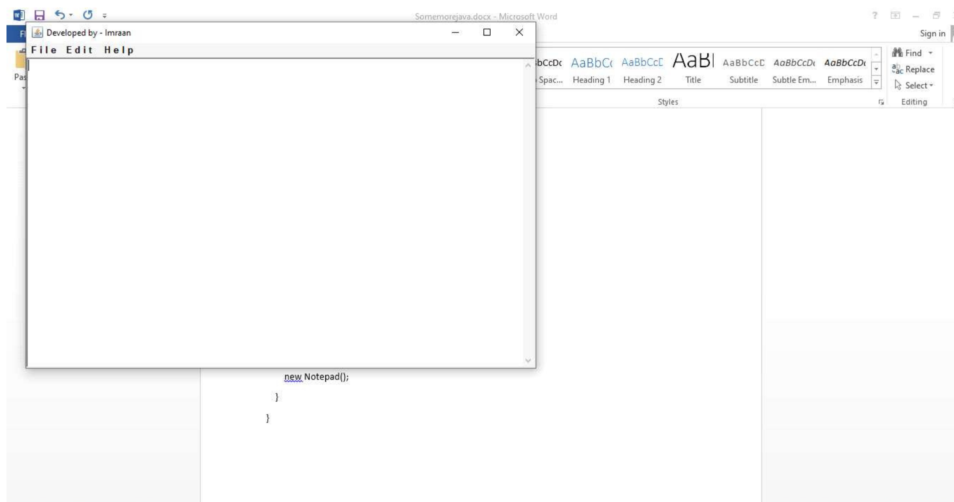
Program

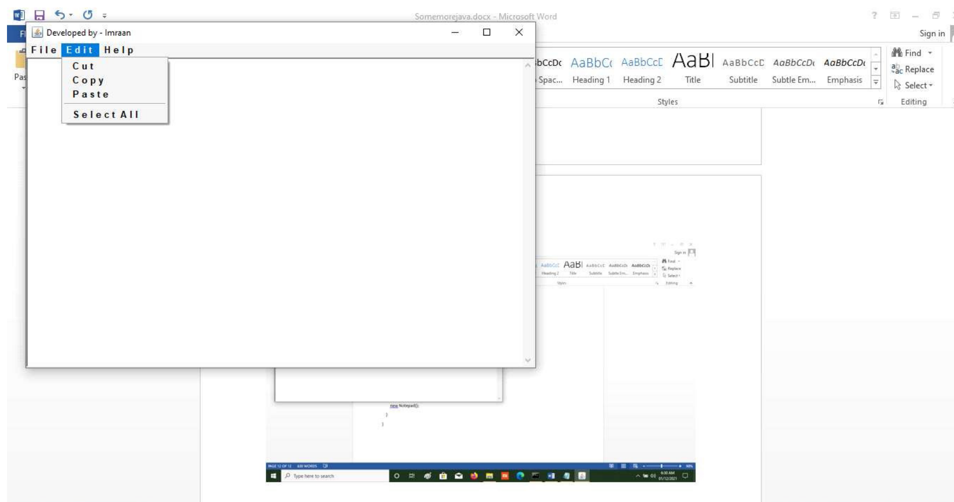
```
importjava.awt.*;
public class Notepad extends Frame
{
   MenuBarmb = new MenuBar();
   Menu f1 = new Menu("F i l e");
   Menu f2 = new Menu("E d i t ");
   Menu f3 = new Menu("H e l p ");
MenuItem a1 = new MenuItem("N e w ");
MenuItem a2 = new MenuItem("O p e n ");
MenuItem a3 = new MenuItem("S a v e ");
MenuItem a4 = new MenuItem("E x i t ");
MenuItem a5 = new MenuItem("C u t ");
MenuItem a6 = new MenuItem("C o p y ");
MenuItem a7 = new MenuItem("P a s t e");
MenuItem a8 = new MenuItem("S e l e c t  A l l ");
MenuItem a9 = new MenuItem("H e l p  T o p i c ");
TextArea ta = new TextArea();
public Notepad()
    {
setTitle("Developed by - Imraan");
setFont(new Font("Courier New",1,13));
      // add item in File menu
f1.add(a1);
f1.add(a2);
f1.add(a3);
f1.addSeparator();
f1.add(a4);
      // add item in edit menu
f2.add(a5);
f2.add(a6);
f2.add(a7);
f2.addSeparator();
f2.add(a8);
      // add item in Help menu
f3.add(a9);
      // add menu in menubar
mb.add(f1);
mb.add(f2);
```

NOTES

```
mb.add(f3);
setMenuBar(mb); // set menubar in frame
add(ta);
setBounds(20,20,740,500); // set Frame bounds
setVisible(true);
    }
public static void main(String[] args)
    {
new Notepad();
    }
}
```

Output of the program:





NOTES

Swing Programs

7. Write a Java Swing program for displaying a list in tree like structure.

Program

```
import javax.swing.*;
import javax.swing.tree.*;
class Swing_Tree_Example
{
public static void main(String[] args)
{
JFrame jf = new JFrame();
DefaultMutableTreeNode d1 = new DefaultMutableTreeNode("Color", true);
DefaultMutableTreeNode d2 = new DefaultMutableTreeNode("Cyan");
DefaultMutableTreeNode d3 = new DefaultMutableTreeNode("Red");
DefaultMutableTreeNode d4 = new DefaultMutableTreeNode("Blue");
DefaultMutableTreeNode d5 = new DefaultMutableTreeNode("Pink");
DefaultMutableTreeNode d6 = new DefaultMutableTreeNode("Green");
DefaultMutableTreeNode d7 = new DefaultMutableTreeNode("White");
d1.add(d2);
d1.add(d3);
d3.add(d4);
d3.add(d5);
d1.add(d6);
d1.add(d7);
JTree jt = new JTree(d1);
jf.add(jt);
jf.setSize(200,200);
jf.setVisible(true);
}
}
```

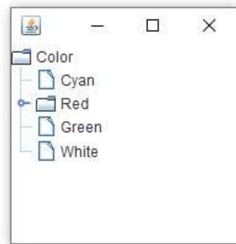
NOTES

Output of the program:

Command Prompt - java Swing_Tree_Example

```
G:\alagappauniv\JavaPrograms>javac Swing_Tree_Example.java

G:\alagappauniv\JavaPrograms>java Swing_Tree_Example
```



8. Write a Java Swing program for displaying numbers in the grid layout.

Program

```
import java.awt.*;
import javax.swing.*;

    public class MyGridLayout{
    JFrame f;
    MyGridLayout(){
        f=new JFrame();

        JButton b1=new JButton("1");
        JButton b2=new JButton("2");
        JButton b3=new JButton("3");
        JButton b4=new JButton("4");
        JButton b5=new JButton("5");
            JButton b6=new JButton("6");
            JButton b7=new JButton("7");
        JButton b8=new JButton("8");
            JButton b9=new JButton("9");

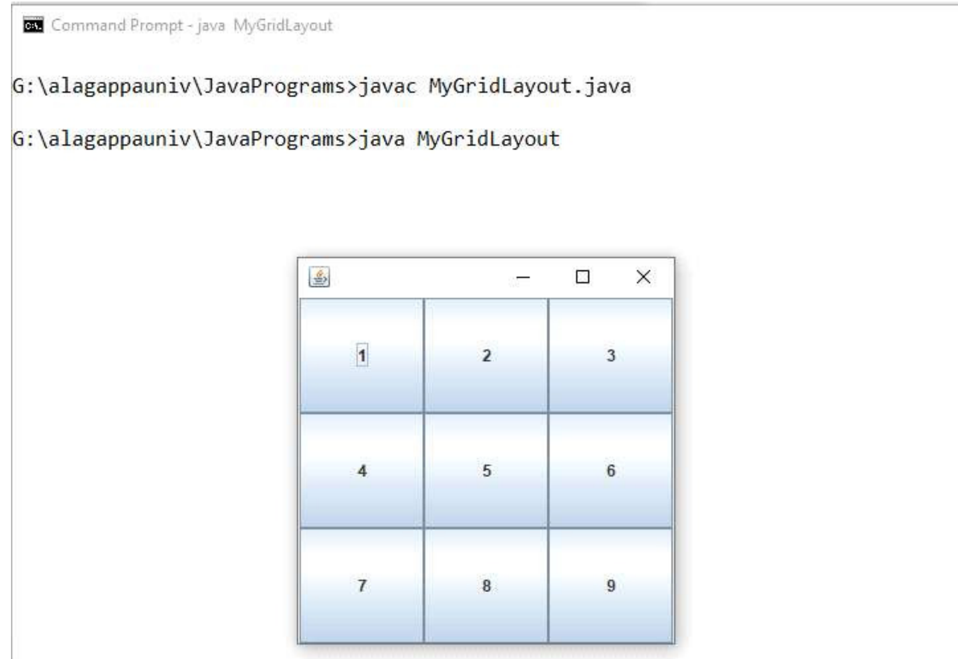

    f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5);
        f.add(b6);f.add(b7);f.add(b8);f.add(b9);
```

NOTES

```
        f.setLayout(new GridLayout(3,3));
        //setting grid layout of 3 rows and 3 columns

        f.setSize(300,300);
        f.setVisible(true);
    }
    public static void main(String[] args) {
        new MyGridLayout();
    }
    }
```

Output of the program:



9. Write a Java Swing program for creating a calculator.

Program

```
import javax.swing.*;
import javax.swing.JOptionPane;
 import java.awt.*;
import java.awt.event.*;
// Class that initialize the applet and create calculator.
public class Calculator extends JApplet
{
public void init()
{
```

NOTES

```
CalculatorPanel calc=new CalculatorPanel();
getContentPane().add(calc);
}}
// Class that creates the calculator panel .
class CalculatorPanel extends JPanel implements
ActionListener
{
// Creation of JButton.
JButton n1,n2,n3,n4,n5,n6,n7,n8,n9,n0, plus, minus, mul,
div,dot,equal;
static JTextField result=new JTextField("0",45);
static String lastCommand=null;
// Create the JObjectPane.
JOptionPane p=new JOptionPane(); double
preRes=0,secVal=0,res;
private static void assign(String no)
{
if((result.getText()).equals("0"))
result.setText(no);
else if(lastCommand=="=")
{
result.setText(no); lastCommand=null; }
else
result.setText(result.getText()+no);
}
// Creation of control panel of calculator and adding
buttons using GridLayout.
public CalculatorPanel()
{ setLayout(new GridLayout());
result.setEditable(false);
result.setSize(300,200);
add(result);
JPanel panel=new JPanel();
panel.setLayout(new GridLayout(5,5));
n7=new JButton("7");
panel.add(n7);
n7.addActionListener(this);
n8=new JButton("8");
panel.add(n8);
n8.addActionListener(this);
n9=new JButton("9");
```

NOTES

```
panel.add(n9);
n9.addActionListener(this);
div=new JButton("/");
panel.add(div);
div.addActionListener(this);
n4=new JButton("4");
panel.add(n4);
n4.addActionListener(this);
n5=new JButton("5");
panel.add(n5);
n5.addActionListener(this);
n6=new JButton("6");
panel.add(n6);
n6.addActionListener(this);
mul=new JButton("*");
panel.add(mul);
mul.addActionListener(this);
n1=new JButton("1");
panel.add(n1);
n1.addActionListener(this);
n2=new JButton("2");
panel.add(n2);
n2.addActionListener(this);
n3=new JButton("3");
panel.add(n3);
n3.addActionListener(this);
minus=new JButton("-");
panel.add(minus);
minus.addActionListener(this);
dot=new JButton(".");
panel.add(dot);
dot.addActionListener(this);
n0=new JButton("0");
panel.add(n0); n0.addActionListener(this);
equal=new JButton("=");
panel.add(equal);
equal.addActionListener(this);
plus=new JButton("+");
panel.add(plus);
plus.addActionListener(this);
add(panel);
```

NOTES

```
}
// Implementing method in ActionListener.
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==n1)
assign("1");
else if(ae.getSource()==n2)
assign("2");
else if(ae.getSource()==n3)
assign("3");
else if(ae.getSource()==n4)
assign("4");
else if(ae.getSource()==n5)
assign("5");
else if(ae.getSource()==n6)
assign("6");
else if(ae.getSource()==n7)
assign("7");
else if(ae.getSource()==n8)
assign("8");
else if(ae.getSource()==n9)
assign("9");
else if(ae.getSource()==n0)
assign("0");
else if(ae.getSource()==dot)
{
if(((result.getText()).indexOf("."))==-1)
result.setText(result.getText()+"."); }
else if(ae.getSource()==minus)
{
preRes=Double.parseDouble(result.getText());
lastCommand="-";
result.setText("0");
}
else if(ae.getSource()==div)
{
preRes=Double.parseDouble(result.getText());
lastCommand="/";
result.setText("0");
}
```

NOTES

```
else if(ae.getSource()==equal)
{
secVal=Double.parseDouble(result.getText());
if(lastCommand.equals("/"))
res=preRes/secVal;
else if(lastCommand.equals("*"))
res=preRes*secVal;
else if(lastCommand.equals("-"))
res=preRes-secVal;
else if(lastCommand.equals("+"))
res=preRes+secVal;
result.setText(" "+res); lastCommand="=";
}
else if(ae.getSource()==mul)
{
preRes=Double.parseDouble(result.getText());
lastCommand="*";
result.setText("0");
}
else if(ae.getSource()==plus)
{
preRes=Double.parseDouble(result.getText());
lastCommand="+";
result.setText("0");
}}
}

/*
<html>
<body>
<applet code="Calculator" width=200 height=300></applet>
</body>
</html>
*/
```

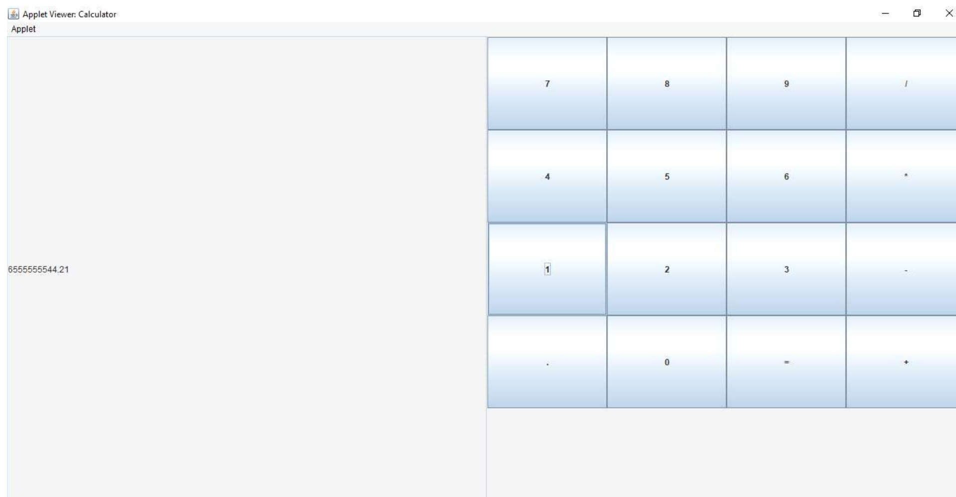
Output of the program:

```
Command Prompt - appletviewer Calculator.java

G:\alagappauniv\JavaPrograms>javac Calculator.java

G:\alagappauniv\JavaPrograms>appletviewer Calculator.java
```

NOTES



10. Write a Java Swing program for creating a Menu.

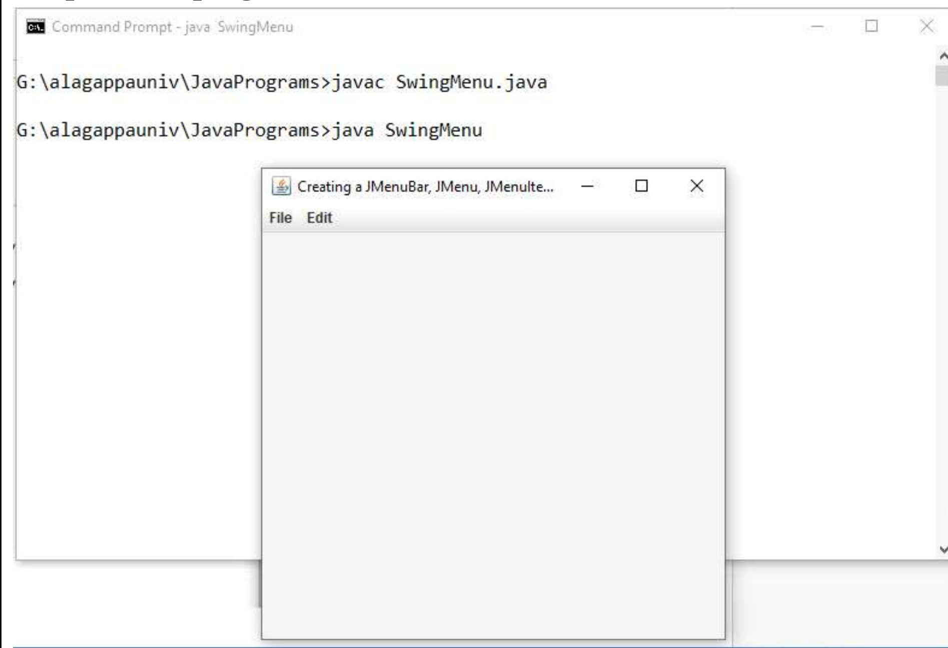
Program

```
import javax.swing.*;
public class SwingMenu
{
public static void main(String[] args)
 {
SwingMenu s = new SwingMenu();
 }
public SwingMenu()
{
JFrame frame = new JFrame("Creating a JMenuBar, JMenu,
JMenuItem and seprator Component");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JMenuBar menubar = new JMenuBar();
JMenu filemenu = new JMenu("File");
filemenu.add(new JSeparator());
JMenu editmenu = new JMenu("Edit");
editmenu.add(new JSeparator());
JMenuItem fileItem1 = new JMenuItem("New");
JMenuItem fileItem2 = new JMenuItem("Open");
JMenuItem fileItem3 = new JMenuItem("Close");
fileItem3.add(new JSeparator());
JMenuItem fileItem4 = new JMenuItem("Save");
JMenuItem editItem1 = new JMenuItem("Cut");
JMenuItem editItem2 = new JMenuItem("Copy");
editItem2.add(new JSeparator());
```

NOTES

```
JMenuItem editItem3 = new JMenuItem("Paste");
JMenuItem editItem4 = new JMenuItem("Insert");
filemenu.add(fileItem1); filemenu.add(fileItem2);
filemenu.add(fileItem3); filemenu.add(fileItem4);
editmenu.add(editItem1);
editmenu.add(editItem2);
editmenu.add(editItem3);
editmenu.add(editItem4);
menubar.add(filemenu);
menubar.add(editmenu);
frame.setJMenuBar(menubar);
frame.setSize(400,400);
frame.setVisible(true);
} }
```

Output of the program:



11. Write a Java program for implementing a colour changer of background with a slider using Swing and Applet.

Program

```
importjavax.swing.*;
importjava.awt.*;
importjavax.swing.event.*;
public class ColorChanger extends JFrame implements
ChangeListener
```

NOTES

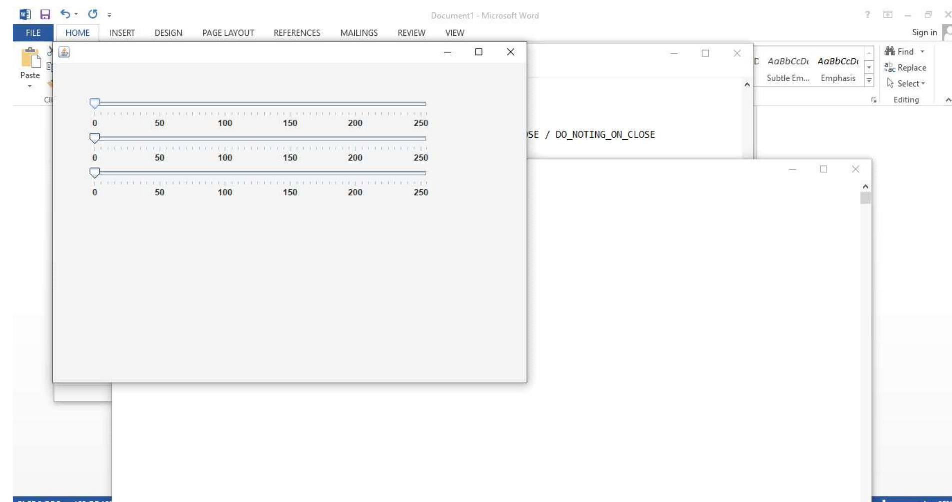
```
{             //type(0-H/1-V),min,max,initial
JSlider r = new JSlider(0,0,255,0);
JSlider g = new JSlider(0,0,255,0);
JSlider b = new JSlider(0,0,255,0);
publicColorChanger()
    {
setLayout(null);
r.setBounds(50,50,500,47);
g.setBounds(50,100,500,47);
b.setBounds(50,150,500,47);
r.setMajorTickSpacing(50);
r.setMinorTickSpacing(5);
r.setPaintTicks(true);
r.setPaintLabels(true);
g.setMajorTickSpacing(50);
g.setMinorTickSpacing(5);
g.setPaintTicks(true);
g.setPaintLabels(true);
b.setMajorTickSpacing(50);
b.setMinorTickSpacing(5);
b.setPaintTicks(true);
b.setPaintLabels(true);

add(r);
add(g);
add(b);
r.addChangeListener(this);
g.addChangeListener(this);
b.addChangeListener(this);
setDefaultCloseOperation(DISPOSE_ON_CLOSE);  //
DISPOSE_ON_CLOSE / HIDE_ON_CLOSE / DO_NOTING_ON_CLOSE
   setBounds(50,50,700,500);
   setVisible(true);
    }
public void stateChanged(ChangeEvent e)
    {
   intcr,cg,cb;
   cr = r.getValue();
   cg = g.getValue();
   cb = b.getValue();
```

NOTES

```
        getContentPane().setBackground(new Color(cr,cg,cb));
        setTitle("Color is -> r = "+cr+" , g = "+cg+" , b =
"+cb);
        }
    public static void main(String[] args)
        {
    newColorChanger();
        }
    }
```

Output of the program:



12. Write a Java program for printing a chessboard pattern using Swing and Applet.

Program

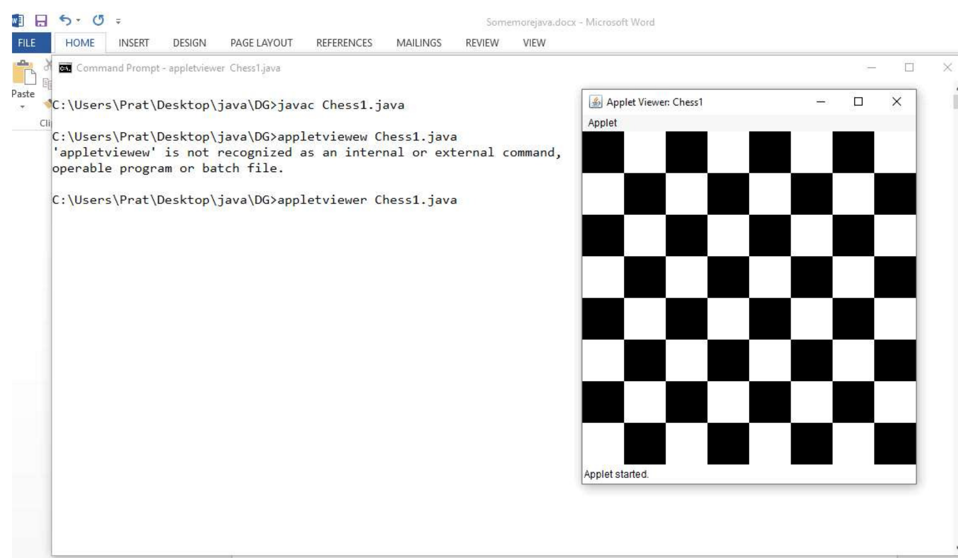
```
import java.applet.*;
import java.awt.*;

/* <applet code="Chess1" width=400 height=400 >
</applet>
*/
public class Chess1 extends Applet
{
int c;
public void paint(Graphics g)
   {
      c = 0;
setBackground(Color.cyan);
for(int y=0;y<400;y+=50)
```

NOTES

```
            {
    c++;
    for(int x=0;x<400;x+=50)
                {
    if(c%2==0)
    g.setColor(Color.white);
    else
    g.setColor(Color.black);
    g.fillRect(x,y,50,50);
    c++;
                }
            }
        }
    }
```

Output of the program:



13. Write a Java program for performing encryption and decryption of a string.

Program

```
    Class Encpt
    {
    public static void main(String aa[])
      {
       String s =”Alagappa University”;
    byte b[] = s.getBytes();
    for(inti=0;i<b.length;i++)
```

NOTES

```
b[i] = (byte)(b[i]-7);   // 7 - encryption key
   s = new String(b);
System.out.println("Encryption Done\t"+""+s);
   b = s.getBytes();
for(inti=0;i<b.length;i++)
b[i] = (byte)(b[i]+7);   // 7 - encryption key
   s = new String(b);
System.out.println("Decryptiion Done , The user had given
the folowing string \t "+ ""+s);


   }
}
```

Output of the program:

```
Command Prompt

C:\Users\Prat\Desktop\java\DG>javac Encpt.java

C:\Users\Prat\Desktop\java\DG>java Encpt
Encryption Done :eZ`ZiiZ□Ngbo^klbmr
Decryptiion Done , The user had given the folowing string        Alagappa University

C:\Users\Prat\Desktop\java\DG>_
```

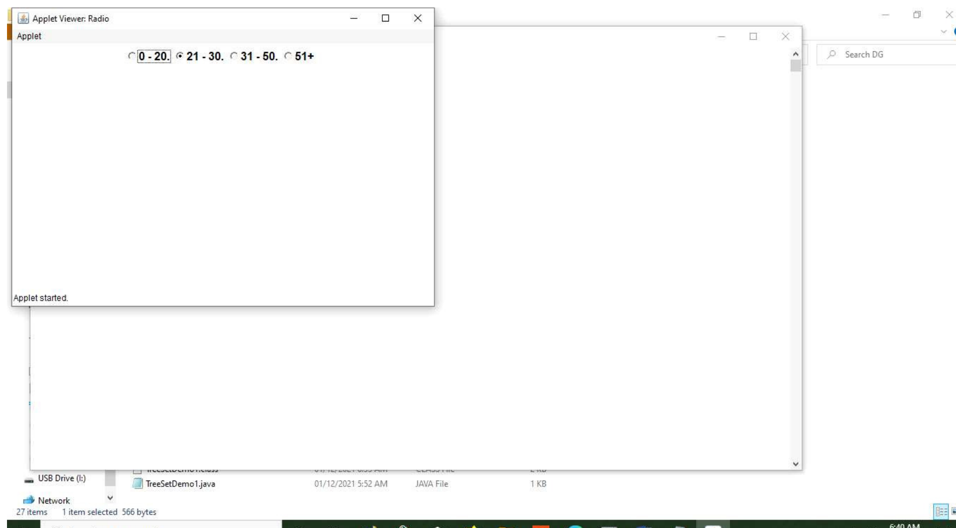
14. Write a Java program to create radio buttons using Swing and Applet.

Program

```
// Radio
import java.applet.*;
import java.awt.*;
public class Radio extends Applet
{
    CheckboxGroup cg = new CheckboxGroup(); // for grouping
    Checkbox c1 = new Checkbox("0 - 20.",cg,false);
    Checkbox c2 = new Checkbox("21 - 30.",cg,true); //
default tick
    Checkbox c3 = new Checkbox("31 - 50.",cg,false);
    Checkbox c4 = new Checkbox("51+ ",cg,false);
    public void init()
     {
setFont(new Font("Courier New",1,16));
//setBackground(Color.RED);
add(c1);
add(c2);
```

```
add(c3);
add(c4);
     }
}
//<applet code=Radio height=350 width=600></applet>
```

Output of the program:



NOTES

15. Write a Java program to show the use of enumerator - enum.

Program

```
enum Color{RED(8),GREEN(10),
    BLUE(16){
    public String getcode(){
       return "#fffddd";
    }
};

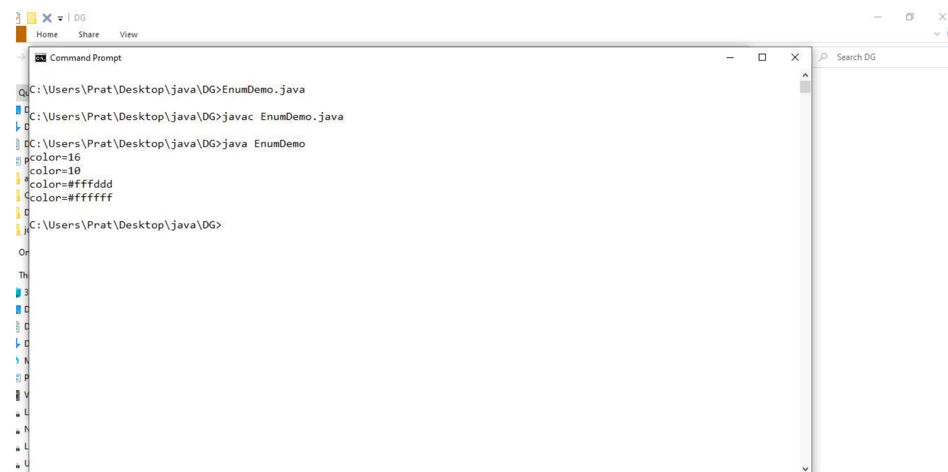
privateint pixel;
Color(int pixel){
this.pixel=pixel;
}
publicintgetPixel(){
    return pixel;
}
public String getcode(){
    return "#ffffff";
}
}
```

NOTES

```
public class EnumDemo {
Color c;
public static void main(String args[]){
EnumDemoed=new EnumDemo();
    ed.c=Color.BLUE;
EnumDemo ed1=new EnumDemo();
    ed1.c=Color.GREEN;
System.out.println("color="+ ed.c.getPixel());
System.out.println("color="+ ed1.c.getPixel());
System.out.println("color="+ ed.c.getcode());
System.out.println("color="+ ed1.c.getcode());

    }
}
```

Output of the program:



16. Write a program on socket programming in Java.

Program

Server.java

```
import java.io.DataInputStream;
import java.net.ServerSocket;
import java.net.Socket;

class server {
    public static void main(String args[]){
        try{
            //create socket,5555 is port number
```

NOTES

```
        ServerSocketserverSocket = new
ServerSocket(5555);

        System.out.println("Waiting for Client to
connect...");

        //establish connection
        Socket socket = serverSocket.accept();
        System.out.println("Client Detected...");
        System.out.println("Client Connected...");

        //fetch incoming message
        DataInputStream dis = new
DataInputStream(socket.getInputStream());
        String  message = (String)dis.readUTF();

        System.out.println("Client message: " + message);

        //close connection
        serverSocket.close();
              }
        catch(Exception e){
        e.printStackTrace();
               }
                    }
                         }
```

Client.java

```
import java.io.DataOutputStream;
import java.net.Socket;
import java.util.Scanner;

class client {
    public static void main(String args[]){
       String message;

       Scanner sc = new Scanner(System.in);

       try{
          //localhost because server is running on local
machine, otherwise use ip of server
```

NOTES

```
                Socket socket = new Socket(“localhost”, 5555);

                System.out.println(“Connected with Server...”);

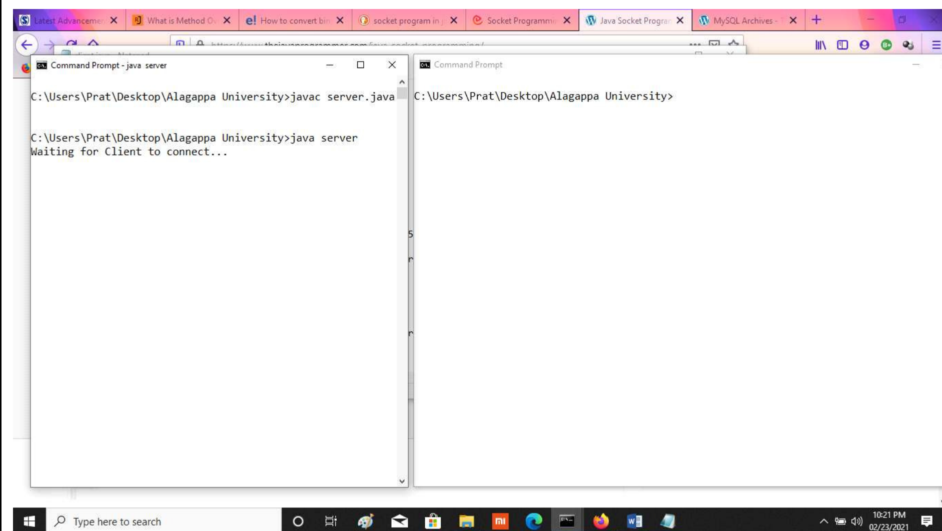
                System.out.println(“enter a message: “);
                message = sc.nextLine();

                //sending the message
                DataOutputStreamdout         =         new
DataOutputStream(socket.getOutputStream());
                dout.writeUTF(message);
                dout.flush();
                dout.close();

                //close connection
                socket.close();
            }catch(Exception e){
                e.printStackTrace();
            }
        }
}
```

Steps to Execute the Program

1. First open a command prompt and run server program. The server will wait for client to be connected.



NOTES

2. Now, open another command prompt and run client program. This will connect client with server. Enter a message at client side to send it to server.

```
C:\Users\Prat\Desktop\Alagappa University>javac server.java

C:\Users\Prat\Desktop\Alagappa University>java server
Waiting for Client to connect...
Client Detected...
Client Connected...
```

```
C:\Users\Prat\Desktop\Alagappa University>javac client.java

C:\Users\Prat\Desktop\Alagappa University>java client
Connected with Server...
enter a message:
```

```
C:\Users\Prat\Desktop\Alagappa University>javac server.java

C:\Users\Prat\Desktop\Alagappa University>java server
Waiting for Client to connect...
Client Detected...
Client Connected...
Client message: Hello , how are you?

C:\Users\Prat\Desktop\Alagappa University>
```

```
C:\Users\Prat\Desktop\Alagappa University>javac client.java

C:\Users\Prat\Desktop\Alagappa University>java client
Connected with Server...
enter a message:
Hello , how are you?

C:\Users\Prat\Desktop\Alagappa University>
```

Try Yourself Questions

1. Write a Java Applet program demonstrating the vehicle moving animation.
2. Write a Java Applet program code to demonstrate that `paint()` method can be called again and again.
3. Write a Java Applet program for displaying a banner "ALAGAPPA UNIVERSITY, KARAIKUDI, TAMIL NADU".
4. Write a Java Applet program for drawing objects and displaying strings.
5. Write a Java Applet program to draw a polygon using `drawPolygon (Polygon p)` function.

NOTES

6. Write a Java program that prints a message "WELCOME TO ALAGAPPA UNIVERSITY" by clicking on the button using AWT Events and Applets.
7. Write a Java Applet program for handling `MouseListener` events.
8. Write a Java Program to illustrate `MouseMotionListener` events.
9. Write a Java Applet program which displays x and y coordinate in its status bar, whenever the user clicks anywhere in the Applet Window.
10. Write a Java Applet program for creating a notepad like application.
11. Write a Java Swing program for displaying numbers in the grid layout.
12. Write a Java Swing program for displaying a list of 10 colours in tree like structure.
13. Write a Java Swing program for creating a simple calculator.
14. Write a Java program for implementing a colour changer of background with a slider using Swing and Applet.
15. Write a Java program for drawing a chessboard pattern using Swing and Applet.
16. Write a Java program to encrypt and decrypt a given string.
17. Write a Java program to demonstrate working on enum in switch case.
18. Write a Java program using socket programming where client sends a text and server receives and prints it.